

1. A method for restoring persistently stored objects of an object oriented environment established on a computer system having a volatile memory and a persistent storage, the method comprising the steps of:

retrieving all segments referenced by said first references and storing them in said volatile memory;

retrieving from said persistent storage a second list containing second references to blocks, that is, portions of said segments, whereby said blocks contain an object description;

creating a new object in said volatile memory and saving the new address of said created object in said second list;

21

determining the new addresses of objects referenced by the newly created object and setting the new address as the reference in the newly created object.

2. The method according to claim 1, whereby said first list and/or said second list are ordered lists.
3. The method according to claim 1 or 2, whereby said first list and/or said second list are organized like a B-tree.
4. The method according to one of the preceding claims, whereby the elements of said first ordered list are indexed by said first references.
5. The method according to one of the preceding claims, whereby each of said first references corresponds to the old memory address at which the respective segment used to reside in the volatile memory.
6. The method according to one of the preceding claims, whereby the elements of said second ordered list are indexed by said second references.
7. The method according to one of the preceding claims, whereby each of said second references corresponds to the old memory address at which the respective block used to reside in said volatile memory.
8. The method according to one of the preceding claims, whereby said object description is formed by a collection of values owned by an object for the variables belonging to its class.



13. The method according to one of the preceding claims, whereby for all references to heads of linked lists the method further comprising the steps of:

reading all blocks of said linked list;

allocating memory to store the value of the variable retrieved from the linked list; and

storing the value in said allocated memory.

14. A method for persistently storing objects of an object oriented environment established on a computer system having a volatile memory and a persistent storage, the method comprising the steps of:

allocating in said volatile memory segments, that is, pieces of memory;

creating a first list (segment map) containing first references to said segments;

creating a second list (object map) containing second references to blocks, that is, portions of said segments;

allocating a block of one of said segments,

creating an object description by saving the values owned by the object of the variables belonging to its class into said allocated block;

adding a new element to said second list containing the particular reference to said created object description;

determining the address of the object description of another object referenced in said object;

setting the address of said respective object description as the reference in the created object description;

storing said second list (object map) on said persistent storage;

storing the segments referenced by said first list (segment map) on said persistent storage; and

storing said first list (segment map) on said persistent storage.

15. The method according to claim 14, whereby said first list and/or said second list are ordered lists.

16. The method according to claim 14 or 15, whereby said first list and/or said second list are organized like a B-tree.

17. The method according to one of the claims 14 to 16, whereby the elements of said first ordered list are indexed by said first references.

18. The method according to one of the claims 14 to 17, whereby each of said first references corresponds to the current memory address at which the respective segment resides in the volatile memory.

19. The method according to one of the claims 14 to 18, whereby the elements of said second ordered list are indexed by said second references.

20. The method according to one of the claims 14 to 19, whereby each of said second references corresponds to the current memory address at which the respective block resides in said volatile memory.

21. The method according to one of the claims 14 to 20, whereby determining the address of the object description of another object referenced in said object comprises the step of searching in said second ordered list (object map) for the element said contains the address of the object description of the referenced other object.

22. The method according to one of the claims 14 to 20, whereby determining the address of the object description of another object referenced in said object comprises the step of using a by a reference to the respective object description provided by each object.

23. The method according to one of the claims 14 to 22, whereby for each value of variables having a variable length the method further comprises the steps of:

allocating a number of portions (blocks) of one of said pieces of memory (segments) that allows to keep the actual value of the variable length variable;

creating a linked list of said number of portions (blocks);

saving value into said number of portions (blocks); and

storing a reference to the head of the linked list in said object description.

24. A computer program product stored on a computer usable medium, comprising computer readable program means for causing a computer to perform a method according to anyone of the preceding claims 1 to 23.